

## Claims:

1. A method for processing a JAVA class to produce an XML document, comprising:  
loading the named JAVA class;

5 determining if the loaded JAVA class implements a predefined interface, said predefined interface comprising annotations including a first parameter, for associating said JAVA class field with a corresponding XML element tag, a second parameter, for specifying a JAVA class to be instantiated when constructing said JAVA class field from said XML file, a third parameter, for identifying a JAVA method to invoke for retrieving  
10 said JAVA class field, and a fourth parameter, for identifying a JAVA method to invoke for retrieving this method; and  
in the case of said loaded JAVA class implementing said predefined interface iteratively processing each field descriptor within the loaded JAVA class to retrieve corresponding XML tag; and

15 transferring field values to new elements created using said corresponding XML tags.

2. A method for processing an XML document to produce a JAVA class, comprising:  
instantiating an object of the desired JAVA class;  
in the case of said instantiated object, implementing a predefined interface,  
iteratively processing each object included within said instantiated object according to the steps of:  
retrieving field descriptors associated with an object being processed;  
creating an object of specified JAVA type for each XML element corresponding to  
25 a field descriptor; and  
storing the created object in the currently processed object.

3. A method for adapting a JAVA object to an application programming interface (API) for converting said JAVA object to eXtensible Markup Language (XML),  
30 comprising:

annotating said JAVA object to include, for each JAVA object to be converted to XML, identification of a respective XML tag, identification of a JAVA class to be instantiated when constructing said JAVA object field from an XML file, identification of a JAVA method to invoke for retrieving said JAVA object and identification of a JAVA method to invoke for retrieving said retrieval method.

5 5 method to invoke for retrieving said retrieval method.

4. An application programming interface (API) for converting at least from JAVA to extensible mark-up language (XML), comprising:

a field description retrieval method, for determining JAVA conversion parameters

10 by examining an annotation associated with each JAVA element to be converted to XML, said annotation defining for each JAVA element at least a corresponding XML tag, a corresponding object class, a corresponding field retrieval method, and a corresponding method retrieval method.

15 5. The API of claim 4, further comprising:

a JAVAtoXML conversion method implemented according to the steps of:

loading the named JAVA class;

determining if the loaded JAVA class implements a predefined interface, said predefined interface comprising annotations including a first parameter, for associating said

20 JAVA class field with a corresponding XML element tag, a second parameter, for specifying a JAVA class to be instantiated when constructing said JAVA class field from said XML file, a third parameter, for identifying a JAVA method to invoke for retrieving said JAVA class field, and a fourth parameter, for identifying a JAVA method to invoke for retrieving this method; and

25 in the case of said loaded JAVA class implementing said predefined interface iteratively processing each field descriptor within the loaded JAVA class to retrieve corresponding XML tag; and

transferring field values to new elements created using said corresponding XML tags.

6. The API of claim 4, further comprising:  
an XMLtoJAVA conversion method, comprising:  
instantiating an object of the desired JAVA class;  
in the case of said instantiated object implementing a predefined interface,  
5 iteratively processing each object included within said instantiated object according to the steps of:  
retrieving field descriptors associated with an object being processed;  
creating an object of specified JAVA type for each XML element corresponding to a field descriptor; and  
10 storing the created object in the currently processed object.

7. A data structure for describing a JAVA class field in a manner facilitating XML conversion, comprising:  
a first parameter, for associating said JAVA class field with a corresponding XML  
15 element tag;  
a second parameter, for specifying a JAVA class to be instantiated when constructing said JAVA class field from said XML file;  
a third parameter, for identifying a JAVA method to invoke for retrieving said JAVA class field; and  
20 a fourth parameter, for identifying a JAVA method to invoke for retrieving this method.

8. The data structure of claim 7, further comprising:  
a fifth parameter, for specifying a type of JAVA object to instantiate for an XML  
25 element representing a collection.

9. The data structure of claim 8, further comprising:  
a sixth parameter, for specifying a tag name to use for each element representing a collection.

10. The data structure of claim 8, wherein said collection comprises a HashTable.

SLA/